

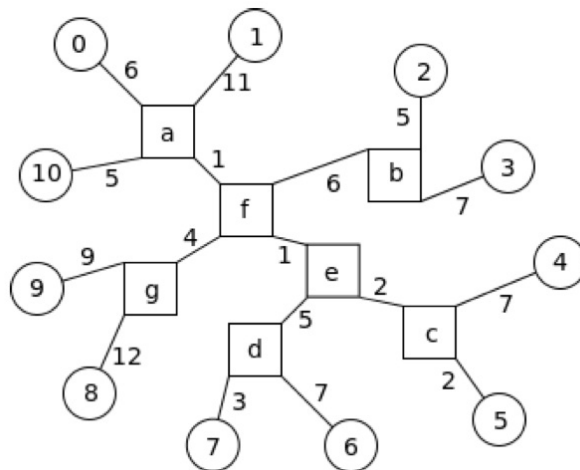
# 도시들

카자흐스탄에는  $N$  개의 소도시가 있고,  $0$  부터  $N - 1$  까지 번호가 붙어 있다. 개수를 정확히 알 수는 없지만, 또 여러개의 대도시가 있다. 소도시와 대도시를 모두 *거주지*라고 부르자.

카자흐스탄의 모든 거주지는 양방향 도로들로 구성된 하나의 도로망으로 연결되어 있다. 각각의 도로는 두개의 서로 다른 거주지를 직접 연결하며, 한 쌍의 거주지는 최대 하나의 도로로 직접 연결되어 있다. 모든 각각의 거주지 쌍  $a, b$ 에 대해  $a$ 에서  $b$ 로 갈 수 있는 유일한 길이 있다(같은 도로를 여러번 사용하지 않는다는 가정 하에).

모든 소도시는 정확히 1개의 다른 거주지와 도로로 직접 연결되어 있고, 모든 대도시는 3개 이상의 거주지와 도로로 직접 연결되어 있다.

다음 그림은 11개의 소도시와 7개의 대도시를 가진 도로망을 보여준다. 소도시는 원으로 표시되어 번호가 붙어 있고, 대도시는 네모로 표시되어 알파벳으로 이름이 붙어 있다.



각각의 도로는 양수 정수인 길이를 가지고 있다. 두 거주지 간의 거리는 한 거주지에서 다른 거주지로 이동하기 위해 사용하는 도로들 길이 합인 최소값이다.

각 대도시  $C$ 에 대해서, 가장 멀리 떨어진 소도시까지의 거리  $r(C)$ 를 생각할 수 있다. 대도시  $C$ 의  $r(C)$ 가 대도시들 중 최소값인 경우  $C$ 를 *허브*라고 부른다. 허브와, 그 허브에서 가장 멀리 떨어진 소도시까지의 거리를  $R$ 로 표시할 것이다. 즉,  $R$ 은 모든  $r(C)$  값들 중 가장 작은 값이다.

위의 예에서, 대도시  $a$ 에서 가장 멀리 떨어진 소도시는 8번이고, 8번까지의 거리는  $r(a) = 1 + 4 + 12 = 17$ 이다. 대도시  $g$ 에 대해서도  $r(g) = 17$ 이다. ( $g$ 에서 가장 멀리 떨어진 소도시 중 하나는 6번이다.) 위의 예에서 허브가 될수 있는 유일한 대도시는  $f$ 이다( $r(f) = 16$ ). 따라서, 위의 예에서  $R = 16$ 이다.

도로망에서 허브를 제거하면, 도로망은 여러개의 연결된 덩어리로 나누어진다. 각 덩어리들이 최대  $\lfloor N/2 \rfloor$ 개의 소도시를 포함하는 경우, 이 허브를 *균형잡힌 것*이라 부른다. (대도시의 수는 세지

않는다는 것에 주의하라.)  $\lfloor x \rfloor$ 라는 표현은  $x$ 보다 크지 않은 자연수들 중에 가장 큰 것을 뜻한다.

위의 예에서  $f$ 는 허브이다. 만약  $f$ 가 제거된다면, 도로망은 4개의 연결된 덩어리로 나누어진다. 4개의 덩어리는 다음과 같다:  $\{0, 1, 10\}$ ,  $\{2, 3\}$ ,  $\{4, 5, 6, 7\}$ ,  $\{8, 9\}$ . 어떤 덩어리에도  $5(= \lfloor 11/2 \rfloor)$  개를 초과하는 소도시가 들어있지 않으므로,  $f$ 는 균형잡힌 허브이다.

## 문제

초기에 당신이 도로망에 대해 알고 있는 정보는 소도시의 수  $N$  뿐이다. 대도시의 수와 도로들이 연결된 방식에 대해서는 알려진 것이 없다. 이들 정보는 소도시의 쌍에 대해서 그 거리를 물어보는 것으로 알아낼 수 밖에 없다.

당신은 다음을 알아내야 한다.

- 모든 부분문제에서: 거리  $R$ .
- 부분문제 3부터 6까지: 균형잡힌 허브가 존재하는지 여부.

hubDistance라는 이름의 함수를 구현해야 한다. 그레이더는 한번의 실행에서 여러개의 테스트 케이스를 사용할 것이다. 한번의 실행에서 사용되는 테스트 케이스의 수는 최대 40개이다. 각 테스트 케이스에 대해서 그레이더는 당신의 hubDistance 함수를 정확히 한번 호출한다. 매번 호출될 때 마다 변수들을 초기화하는 것을 잊지 말도록 하라.

- hubDistance(N, sub)
  - N: 소도시의 수.
  - sub: 부분문제 번호 (부분문제 절에서 설명됨).
  - sub가 1 혹은 2인 경우, 함수는  $R$ 이나  $-R$ 을 리턴하면 된다.
  - sub가 2보다 큰 경우, 함수는, 균형잡힌 허브가 존재하는 경우  $R$ 을, 존재하지 않는 경우  $-R$ 을 리턴하여야 한다.

hubDistance가 실행될 때 그레이더 함수 getDistance(i, j)를 호출하여 도로망에 대한 정보를 얻을 수 있다. 이 함수는 소도시  $i$ 와  $j$  간의 거리를 리턴한다. 만약  $i$ 와  $j$ 가 같다면 0이 리턴된다는 것을 주의하라. 잘못된 인자가 주어진 경우에도 0이 리턴된다.

## 부분문제

각 테스트 케이스에서,

- $N$ 은 6 이상 110 이하이다.
- 임의의 두 소도시 간의 거리는 1 이상 1,000,000 이하이다.

당신의 프로그램이 사용할 수 있는 질의의 수는 제한되어 있다. 이 제한은 부분문제에 따라 다르며 아래 표에 주어져 있다. 만약 당신의 프로그램이 이 제한을 넘어설 경우, 실행은 종료되며 잘못된 답이 나온 것으로 간주된다.

부분 문제	점수	질의 개수	균형잡힌 허브 찾기 여부	추가제한
1	13	$\frac{N(N-1)}{2}$	NO	없음
2	12	$\lceil \frac{7N}{2} \rceil$	NO	없음
3	13	$\frac{N(N-1)}{2}$	YES	없음
4	10	$\lceil \frac{7N}{2} \rceil$	YES	모든 대도시는 정확히 세개의 거주지와 도로로 직접 연결되어 있음
5	13	$5N$	YES	없음
6	39	$\lceil \frac{7N}{2} \rceil$	YES	없음

$\lceil x \rceil$ 라는 표현은  $x$ 보다 크거나 같은 자연수들 중에 가장 작은 것을 뜻함에 주의하라.

### Sample grader

부분문제 번호가 입력에 주어짐에 주의하라. Sample grader는 부분문제 번호가 무엇이냐에 따라 다르게 동작한다.

Sample grader는 입력파일 towns.in에서 입력을 읽는다. 양식은 아래와 같다.

- line 1: 부분문제 번호와 테스트 케이스의 개수
- line 2:  $N_1$ , 첫번째 테스트 케이스에서 소도시의 개수.
- 다음  $N_1$ 개의 줄: 이 줄들 중  $i$  ( $1 \leq i \leq N_1$ )번째 줄의  $j$  ( $1 \leq j \leq N_1$ )번째 수는 소도시  $i - 1$ 과  $j - 1$  간의 거리이다.
- 첫번째 테스트 케이스와 동일한 양식으로 나머지 테스트 케이스들이 주어진다.

각 테스트 케이스에 대해서 sample grader는 hubDistance의 리턴 값과 질의 호출의 개수를 두 줄에 출력한다.

위의 예제에 해당하는 입력파일의 내용은 아래와 같다.

```

1 1
11
0 17 18 20 17 12 20 16 23 20 11
17 0 23 25 22 17 25 21 28 25 16
18 23 0 12 21 16 24 20 27 24 17
20 25 12 0 23 18 26 22 29 26 19
17 22 21 23 0 9 21 17 26 23 16
12 17 16 18 9 0 16 12 21 18 11
20 25 24 26 21 16 0 10 29 26 19
16 21 20 22 17 12 10 0 25 22 15
23 28 27 29 26 21 29 25 0 21 22
20 25 24 26 23 18 26 22 21 0 19
11 16 17 19 16 11 19 15 22 19 0

```

이 입력 양식은 도로들을 제시하는 것과는 많이 다르다. Sample grader를 변형하여 다른 입력 양식을 사용하도록 하고 실험하는 것은 허용된다.